

AD-A181 959

INTEGRATED INFORMATION SUPPORT SYSTEM (IISS) VOLUME 5

1/1

COMMON DATA MODEL S. (U) GENERAL ELECTRIC CO

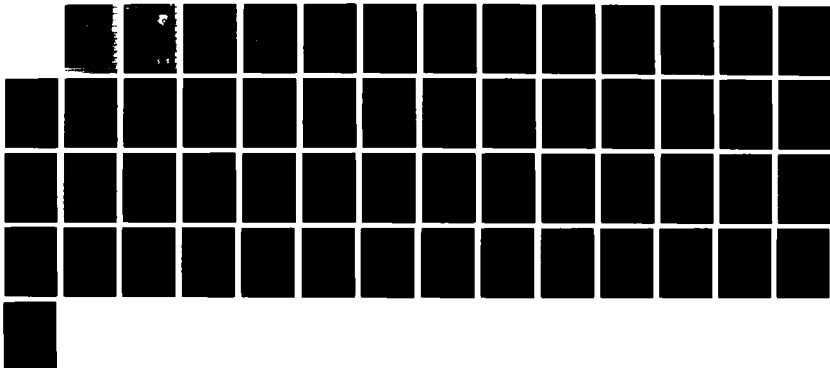
SCHENECTADY NY PRODUCTION RESOURCES CONSU.

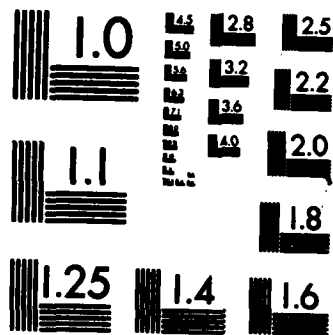
UNCLASSIFIED

J L ALTHOFF ET AL. 01 NOV 85 P5-620141213

F/G 12/5

NL





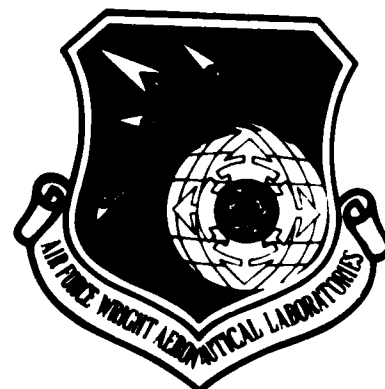
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

DTIC FILE COPY

2

**AFVAL-TR-86-4006
Volume V
Part 13**

AD-A181 959



**INTEGRATED INFORMATION
SUPPORT SYSTEM (IISS)
Volume V - Common Data Model Subsystem
Part 13 - NDML Precompiler Parse NDML
Product Specification**

**General Electric Company
Production Resources Consulting
One River Road
Schenectady, New York 12345**

**Final Report for Period 22 September 1980 - 31 July 1985
November 1985**

Approved for public release; distribution is unlimited.

PREPARED FOR:

**MATERIALS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AFB, OH 45433-6533**

**DTIC
ELECTE
JUL 01 1987
S D E**

87 6 29 001

Unclassified

1 November 1985

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

A181 959

1a. REPORT SECURITY CLASSIFICATION Unclassified		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			
4. PERFORMING ORGANIZATION REPORT NUMBER(S)		5. MONITORING ORGANIZATION REPORT NUMBER(S) AFVAL-TR-86-4006 Vol V, Part 13	
6a. NAME OF PERFORMING ORGANIZATION General Electric Company Production Resources Consulting	6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION AFVAL/MLTC	
6c. ADDRESS (City, State and ZIP Code) 1 River Road Schenectady, NY 12345		7b. ADDRESS (City, State and ZIP Code) WPAFB, OH 45433-6533	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Materials Laboratory Air Force Systems Command, USAF	8b. OFFICE SYMBOL (If applicable) AFVAL/MLTC	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F33619-80-C-5185	
8c. ADDRESS (City, State and ZIP Code) Wright-Patterson AFB, Ohio 45433		10. SOURCE OF FUNDING NOS.	
		PROGRAM ELEMENT NO. 78011F	TASK NO. 62
		PROJECT NO. 7500	WORK UNIT NO. 01
11. TITLE (Include Security Classification) (See Reverse)			
12. PERSONAL AUTHOR(S) Althoff, J. L. and Singh, S.			
13a. TYPE OF REPORT Final Technical Report	13b. TIME COVERED 22 Sept 1980 - 31 July 1985	14. DATE OF REPORT (Yr., Mo., Day) 1985 November	15. PAGE COUNT 55
16. SUPPLEMENTARY NOTATION ICAM Project Priority 6201 The computer software contained herein are theoretical and/or references that in no way reflect Air Force-owned or -developed computer software.			
17. ESOAT CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB GR.	
1808	0005		
19. ABSTRACT (Continue on reverse if necessary and identify by block number) This document is the product specification establishing the design implementation of the IISS Configuration Item PRE3 which will parse and check Neutral Data Manipulation Language (NDML) statements for correct syntax. → See p I-1			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS <input type="checkbox"/>		21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL David L. Judson		22b. TELEPHONE NUMBER (Include Area Code) 813-255-0976	22c. OFFICE SYMBOL AFVAL/MLTC

11. Title

Integrated Information Support System (IISS)
Vol V - Common Data Model Subsystem
Part 13 - NDML Precompiler Parse NDML
Product Specification

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



PREFACE

This product specification covers the work performed under Air Force Contract F33615-80-C-5155 (ICAM Project 6201). This contract is sponsored by the Materials Laboratory, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Gerald C. Shumaker, ICAM Program Manager, Manufacturing Technology Division, through Project Manager, Mr. David Judson. The Prime Contractor was Production Resources Consulting of the General Electric Company, Schenectady, New York, under the direction of Mr. Alan Rubenstein. The General Electric Project Manager was Mr. Myron Hurlbut of Industrial Automation Systems Department, Albany, New York.

Certain work aimed at improving Test Bed Technology has been performed by other contracts with Project 6201 performing integrating functions. This work consisted of enhancements to Test Bed software and establishment and operation of Test Bed hardware and communications for developers and other users. Documentation relating to the Test Bed from all of these contractors and projects have been integrated under Project 6201 for publication and treatment as an integrated set of documents. The particular contributors to each document are noted on the Report Documentation Page (DD1473). A listing and description of the entire project documentation system and how they are related is contained in document FTR620100001, Project Overview.

The subcontractors and their contributing activities were as follows:

TASK 4.2

<u>Subcontractors</u>	<u>Role</u>
Boeing Military Aircraft Company (BMAC)	Reviewer
D. Appleton Company (DACOM)	Responsible for IDEF support, state-of-the-art literature search
General Dynamics/ Ft. Worth	Responsible for factory view function and information models

PS 620141213
1 November 1985

Subcontractors

Role

Illinois Institute of
Technology

Responsible for factory view
function research (IITRI)
and information models of
small and medium-size business

North American Rockwell

Reviewer

Northrop Corporation

Responsible for factory view
function and information
models

Pritsker and Associates

Responsible for IDEF2 support

SofTech

Responsible for IDEF0 support

TASKS 4.3 - 4.9 (TEST BED)

Subcontractors

Role

Boeing Military Aircraft
Company (BMAC)

Responsible for consultation on
applications of the technology
and on IBM computer technology.

Computer Technology
Associates (CTA)

Assisted in the areas of
communications systems, system
design and integration
methodology, and design of the
Network Transaction Manager.

Control Data Corporation
(CDC)

Responsible for the Common Data
Model (CDM) implementation and
part of the CDM design (shared
with DACOM).

D. Appleton Company
(DACOM)

Responsible for the overall CDM
Subsystem design integration
and test plan, as well as part
of the design of the CDM
(shared with CDC). DACOM also
developed the Integration
Methodology and did the schema
mappings for the Application
Subsystems.

PS 620141213
1 November 1985

Subcontractors

Role

Digital Equipment
Corporation (DEC)

Consulting and support of the
performance testing and on DEC
software and computer systems
operation.

McDonnell Douglas
Automation Company
(McAuto)

Responsible for the support and
enhancements to the Network
Transaction Manager Subsystem
during 1984/1985 period.

On-Line Software
International (OSI)

Responsible for programming the
Communications Subsystem on the
IBM and for consulting on the
IBM.

Rath and Strong Systems
Products (RSSP) (In 1985
became McCormack & Dodge)

Responsible for assistance in
the implementation and use of
the MRP II package (PIOS) that
they supplied.

SofTech, Inc.

Responsible for the design and
implementation of the Network
Transaction Manager (NTM) in
1981/1984 period.

Software Performance
Engineering (SPE)

Responsible for directing the
work on performance evaluation
and analysis.

Structural Dynamics
Research Corporation
(SDRC)

Responsible for the User
Interface and Virtual Terminal
Interface Subsystems.

Other prime contractors under other projects who have
contributed to Test Bed Technology, their contributing
activities and responsible projects are as follows:

<u>Contractors</u>	<u>ICAM Project</u>	<u>Contributing Activities</u>
Boeing Military Aircraft Company (BMAC)	1701, 2201, 2202	Enhancements for IBM node use. Technology Transfer to Integrated Sheet Metal Center (ISMC)

PS 620141213
1 November 1985

<u>Contractors</u>	<u>ICAM Project</u>	<u>Contributing Activities</u>
Control Data Corporation (CDC)	1502, 1701	IISS enhancements to Common Data Model Processor (CDMP)
D. Appleton Company (DACOM)	1502	IISS enhancements to Integration Methodology
General Electric	1502	Operation of the Test Bed and communications equipment.
Hughes Aircraft Company (HAC)	1701	Test Bed enhancements
Structural Dynamics Research Corporation (SDRC)	1502, 1701, 1703	IISS enhancements to User Interface/Virtual Terminal Interface (UI/VTI)
Systran	1502	Test Bed enhancements. Operation of Test Bed.

TABLE OF CONTENTS

		<u>Page</u>
SECTION 1.0	SCOPE	1-1
1.1	Identification	1-1
1.2	Functional Summary	1-1
SECTION 2.0	DOCUMENTS	2-1
2.1	Reference Documents	2-1
2.2	Terms and Abbreviations	2-1
SECTION 3.0	REQUIREMENTS	3-1
3.1	Structural Description	3-1
3.2	Functional Flow	3-1
3.3	Interfaces	3-2
3.3.1	Inputs/Outputs	3-2
3.4	Program Interrupts	3-2
3.5	Timing and Sequencing Description ...	3-2
3.6	Special Control Features	3-3
3.7	Storage Allocation	3-3
3.7.1	Database Definition	3-3
3.7.1.1	File Description	3-3
3.7.1.2	Table Description	3-3
3.7.1.3	Item Description	3-3
3.8	Object Code Creation	3-3
3.9	Adaptation Data	3-3
3.10	Detail Design Description	3-3
3.10.1	Main Program List	3-4
3.10.2	Module List	3-6
3.10.3	External Routines List	3-8
3.10.4	Include File List	3-10
3.10.5	Where Include File Used List	3-12
3.10.6	Where External Routine Used List	3-15
3.10.7	Main Program Parts List	3-18
3.10.8	Module Documentation	3-2.
3.10.9	Include File Descriptions	3-28
3.10.10	Hierarchy Chart	3-36
3.11	Program Listings Comments	3-41
SECTION 4.0	QUALITY ASSURANCE PROVISIONS	4-1
4.1	Introduction and Definitions	4-1
4.2	Computer Programming Test and Evaluation	4-1

SECTION 1

SCOPE

1.1 Identification

This specification establishes the design of Function PRE3, Parse NDML, one of the major functions of the Configuration Item "Precompiler" to be built and formally accepted by the ICAM Program Office. This CPCI constitutes one of the subsystems of the Common Data Model Processor (CDMP). A

1.2 Functional Summary

This function parses the NDML statements into tokenized form. Legal syntax is checked and syntax error messages issued.

Keywords: ICAM (Integrated Computer Aided Manufacturing);
CPCI (Computer Program Configuration Item).

SECTION 2

DOCUMENTS

2.1 Reference Documents

1. ICAM Documentation Standards: IDS15012000A, 28 December 1981.
2. D. Appleton Co., CDM Administrators Manual: UM620141000, March 1984.
3. D. Appleton Co., CDM1-IDEF1 Model of the Common Data Model; CCS620141000, 15 May 1985.
4. D. Appleton Co., Computer Program Development Specification (DS) for ICAM Integrated Support System (IISS) Configuration Item: NDML Precompiler; DS620141200, October 1984.
5. D. Appleton Co., Embedded NDML Programmer's Reference Manual; PRM620141200, March 1985.
6. Softech, Inc., NTM Programmer's Guide: UM620140001, July 1984.
7. Control Data Corporation, Computer Program Development Specification (DS) for ICAM Integrated Support System (IISS) Configuration Item: NDDL Command Processor; DS620141100, June 1985.

2.2 Terms and Abbreviations

Attribute Use Class: (AUC)

Conceptual Schema: (CS)

Common Data Model Processor: (CDMP)

Common Data Model: (CDM) Describes common data application process formats, form definitions, etc, of the IISS and includes conceptual schema, external, internal schemas, and schema transformation operators.

Data Field: (DF) An element of data in the external

schema. It is by this name that an NDML programmer references data.

Database Management System: (DBMS)

Distributed Request Supervisor: (DRS) This IISS CDM subsystem configuration item controls the execution of distributed NDML queries and non distributed updates.

Domain: A logical definition of legal attribute class values.

Domain Constraint: Predicate that applies to a single domain.

External Schema: (ES)

Forms: Structured views which may be imposed on windows or other forms. A form is composed of fields where each field is a form, item, or window.

Forms Processor: (FP) A set of callable execution time routines available to an application program for form processing.

Internal Schema: (IS)

Integrated Information Support System: (IISS) A test computing environment used to investigate, demonstrate and test the concepts of information management and information integration in the context of Aerospace Manufacturing. The IISS addresses the problems of integration of data resident on heterogeneous databases supported by heterogeneous computers interconnected via a local Area Network.

Mapping: The correspondence of independent objects in two schemas: ES to CS or CS to IS.

Network Transaction Manager: (NTM) Performs the coordination, communication and housekeeping functions required to integrate the application processes and system services resident on the various hosts into a cohesive system.

Neutral Data Manipulation Language: (NDML) A language developed by the IISS project to provide uniform access to common data, regardless of database manager or distribution criteria. It provides distributed retrieved and single node

updates.

ORACLE: Relational DBMS based on the SQL (Structured Query Language, a product of ORACLE Corp, Menlo Park, CA). The CDM is an ORACLE database.

Parcel: A sequential file containing sections source code of the input application program.

Request Processor: (RP) A COBOL program that will satisfy a retrieval or update NDML subtransaction against a particular Database Management System.

User Interface: (UI) Controls the user's terminal and interfaces with the rest of the system.

Virtual Terminal Interface: (VTI) Performs the interfacing between different terminals and the UI. This is done by defining a specific set of terminal features and protocols which must be supported by UI software which constitutes the Virtual Terminal Definition. Specific terminals are then mapped against the Virtual Terminal software by specific software modules written for each type of real terminal supported.

SECTION 3

REQUIREMENTS

3.1 Structural Description

A graphic portrayal of this Computer Program Configuration Item (CPCI) is included in Section 3.10. This chart shows the hierarchical relationship of each module making up this CPCI. This module is accessed for each NDML command identified by PRE2. The routine is generated by the UNIX tools YACC and LEX. YACC generates a parser given the syntax rules as input. LEX generates the lexical analyzer for the language given the definitions of the lexical units of the language. The parser generated by YACC is named YYPARSE and the lexical analyzer is named YYLEX. PRE2 controls the call to YYPARSE. Whenever YYPARSE (the parser) needs another token or unit of command input, YYLEX is called. Whenever YYLEX needs another character of the user command, YYINPUT is called. In this application, YYINPUT has been modified to record the next character from the file containing NDML commands. The parsed tokens are stored in global data structures accessible by "C" primitives. PRE2 then uses these primitives to access the "C" data structures, storing them in COBOL tables and checking command semantics.

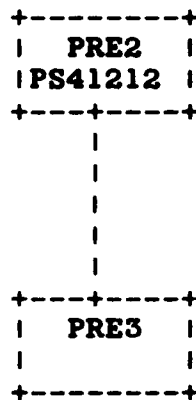
3.2 Functional Flow

This CPCI implements the logic defined in the Development Specification for this CPCI. Details of inputs/outputs and relationships between modules are to be found in Section 3.10.

This CPCI has been designated to operate in a batch or interactive mode. It must operate in the system environment established for IISS; that is, use of the Network Transaction Manager. It must use the ORACLE DBMS installed on a DEC VAX computer.

3.3 Interfaces

The following diagram depicts the interface of PRE3 with other CPCI's in the system.



3.3.1 Inputs/Outputs

The following table depicts the inputs and outputs of this CPCI. A detailed description for each item can be found in the DS for this CPCI.

FUNCTION: PRE3

INPUT	OUTPUT
Command Number	
Module Status	

3.4 Program Interrupts

Not applicable to this CPCI.

3.5 Timing and Sequencing Description

PRE3 is called once for every NDML statement encountered in the user's input program.

3.6 Special Control Features

Not applicable to this CPCI.

3.7 Storage Allocation

3.7.1 Database Definition

No databases are used by this CPCI.

3.7.1.1 File Description

No permanent files have been defined for this CPCI. It may use temporary scratch files for such things as generated program source code or temporary query results.

3.7.1.2 Table Description

All tables used by this CPCI have been defined by the Development Specification for this CPCI.

3.7.1.3 Item Description

Not applicable to this CPCI.

3.8 Object Code Creation

The object code for this CPCI will be created by the system integration test team by using defined IISS Software Configuration Management procedures. This CPCI will use the COBOL and "C" language compilers. In addition, source code generated by the YACC and LEX tools must be compiled to become objects in the final, executable software package.

3.9 Adaptation Data

This CPCI has been coded using ANSI COBOL and a "standard" subset of the "C" languages. The intent was to provide a transportable system. Any system environment supporting these languages, a virtual memory management scheme, the COMM and NTM subsystems of IISS and the ORACLE Database Management System should be able to support this CPCI. Every possible attempt has been made to localize and identify any machine or environment dependent modules through the original design of the IISS and application of Configuration Management Procedures.

3.10 Detail Design Description

The following sections have been computer generated for this CPCI.

3.10.1 Main Program List

The following is a list of all "Main Programs" which are modules that are not called by any other module being documented here. These modules are either program entry points or, if they are hooked into another set of programs via subroutine calls, they are the points the external programs can call and therefore enter through. To differentiate between the two types of entry points, look at the individual Module Documentation (section 3.10.8) and look at Module Type for each of the Main Program modules listed. Note whether the routine is a Program, Subroutine, or Function. If it is a Program, it is truly a main program entry point. If not, then it is merely called by other programs not being documented here.

PS 620141213
1 November 1985

PARSE NDML COMMAND Main Program List

Module Name -----	Purpose -----
NDMLPAR	**** PURPOSE NOT FOUND BY STRIPPER ****
YYPARSE	**** PURPOSE NOT FOUND BY STRIPPER ****
YYWRAP	**** PURPOSE NOT FOUND BY STRIPPER ****

PS 620141213
1 November 1985

3.10.2 Module List

The following is a list of all the modules being documented here along with their purpose. Each module has a unique name, no matter what language it was written in.

PS 620141213
1 November 1985

PARSE NDML COMMAND Module List

Module Name -----	Purpose -----
NDMLPAR	**** PURPOSE NOT FOUND BY STRIPPER ****
YYERROR	**** PURPOSE NOT FOUND BY STRIPPER ****
YYPARSE	**** PURPOSE NOT FOUND BY STRIPPER ****
YYWRAP	**** PURPOSE NOT FOUND BY STRIPPER ****

PS 620141213
1 November 1985

3.10.3 External Routines List

The following is a list of all routines or functions not documented here that are called by modules that are documented here. The first caller, in alphabetical order, is listed as well. The specification in which any module is documented may be found in the Module Documentation Index (Document Number CM 620100001). See section 3.10.6 for a list of the modules that call each of these external routines.

PS 620141213
1 November 1985

PARSE NDML COMMAND External Routines List

Module Name	First User
-----	-----
ADDNOCORR	YYPARSE
ADD_CORR	YYPARSE
ADD_TO_CMT	YYPARSE
ADD_TO_LST	YYPARSE
INITCMD	NDMLPAR
NDMLPAS	NDMLPAR
PRINTF	YYPARSE
SPRINTF	YYERROR
STRLEN	YYERROR
WRITERR	YYERROR
YYINPUT	NDMLPAR
YYLEX	YYPARSE
YYOUTPUT	NDMLPAR

3.10.4 Include File List

The following is a list of all include files called in by modules being documented here. Each include file has a unique name regardless of the language being used. The purpose of each include file is listed as well. A more complete description of each include file is given in section 3.10.9. The purpose listed is the one that is in the source code of the include file.

A purpose of "**** PURPOSE NOT FOUND BY STRIPPER ****" indicates that a purpose statement was not written into the include file itself. The most common reason for this is that the include file comes from system libraries that were not developed by the project, such as 'C' libraries that are provided with the 'C' compiler.

See section 3.10.6 for a set of lists which show all the modules which call in each of these include files.

PS 620141213
1 November 1985

PARSE NDML COMMAND Include File List

File Name -----	Purpose -----
CMD5	COMMAND NO. FOR EACH NDDL/NDML COMMAND
LISTID	PROVIDES LIST OF PARSED OBJECTS
LISTS	PROVIDES THE DIMENSIONS OF THE NDDL LISTS
NDMLEXY	LEX GENERATED INCLUDE FILE
NDMLYAC.INP"	**** PURPOSE NOT FOUND BY STRIPPER ****
STDIO	**** PURPOSE NOT FOUND BY STRIPPER ****
STDTP	STANDARD TYPE DEFINITIONS

PS 620141213
1 November 1985

3.10.5 Where Include File Used List

The following lists each include file from 3.10.4 and all the modules documented in this specification which include them. The purpose of each module is listed as well.

PS 620141213
1 November 1985

PARSE NDML COMMAND Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
CMDS		
	YYERROR	**** PURPOSE NOT FOUND BY STRIPPER ****
	YYPARSE	**** PURPOSE NOT FOUND BY STRIPPER ****
	YYWRAP	**** PURPOSE NOT FOUND BY STRIPPER ****
LISTID		
	YYERROR	**** PURPOSE NOT FOUND BY STRIPPER ****
	YYPARSE	**** PURPOSE NOT FOUND BY STRIPPER ****
	YYWRAP	**** PURPOSE NOT FOUND BY STRIPPER ****
LISTS		
	YYERROR	**** PURPOSE NOT FOUND BY STRIPPER ****
	YYPARSE	**** PURPOSE NOT FOUND BY STRIPPER ****
	YYWRAP	**** PURPOSE NOT FOUND BY STRIPPER ****
NDMLEXY		
	YYERROR	**** PURPOSE NOT FOUND BY STRIPPER ****
	YYPARSE	**** PURPOSE NOT FOUND BY STRIPPER ****
	YYWRAP	**** PURPOSE NOT FOUND BY STRIPPER ****
NDMLYAC.INP"		
	YYERROR	**** PURPOSE NOT FOUND BY STRIPPER ****
	YYPARSE	**** PURPOSE NOT FOUND BY STRIPPER ****
	YYWRAP	**** PURPOSE NOT FOUND BY STRIPPER ****

PS 620141213
1 November 1985

PARSE NDML COMMAND Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
STDIO		
	NDMLPAR	**** PURPOSE NOT FOUND BY STRIPPER ****
	YYERROR	**** PURPOSE NOT FOUND BY STRIPPER ****
	YYPARSE	**** PURPOSE NOT FOUND BY STRIPPER ****
	YYWRAP	**** PURPOSE NOT FOUND BY STRIPPER ****
STDYTP		
	NDMLPAR	**** PURPOSE NOT FOUND BY STRIPPER ****

PS 620141213
1 November 1985

3.10.6 Where External Routine Used List

The following lists each external function or routine listed in 3.10.3 and all the documented modules which call it. The purpose of each module is listed as well.

PS 620141213
1 November 1985

PARSE NDML COMMAND Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
ADDNCORR	YYPARSE	**** PURPOSE NOT FOUND BY STRIPPER ****
ADD_CORR	YYPARSE	**** PURPOSE NOT FOUND BY STRIPPER ****
ADD_TO_CNT	YYPARSE	**** PURPOSE NOT FOUND BY STRIPPER ****
ADD_TO_LST	YYPARSE	**** PURPOSE NOT FOUND BY STRIPPER ****
INITCMD	NDMLPAR	**** PURPOSE NOT FOUND BY STRIPPER ****
NDMLPAS	NDMLPAR	**** PURPOSE NOT FOUND BY STRIPPER ****
PRINTF	YYPARSE	**** PURPOSE NOT FOUND BY STRIPPER ****
SPRINTF	YYERROR	**** PURPOSE NOT FOUND BY STRIPPER ****

PS 620141213
1 November 1985

PARSE NDML COMMAND Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
STRLEN	YYERROR	**** PURPOSE NOT FOUND BY STRIPPER ****
	YYPARSE	**** PURPOSE NOT FOUND BY STRIPPER ****
WRITERR	YYERROR	**** PURPOSE NOT FOUND BY STRIPPER ****
YYINPUT	NDMLPAR	**** PURPOSE NOT FOUND BY STRIPPER ****
YYLEX	YYPARSE	**** PURPOSE NOT FOUND BY STRIPPER ****
YYOUTPUT	NDMLPAR	**** PURPOSE NOT FOUND BY STRIPPER ****

PS 620141213
1 November 1985

3.10.7 Main Program Parts List

The following lists each Main Program listed in 3.10.1 and all the modules which are called either by that module itself or by any of the documented modules which it calls. It is possible for a non-main module to be listed more than once if it is called by multiple modules. The called modules, in this case known as program parts, are marked as to whether they are documented here. If so, the phrase "well-defined module" appears by the module name, if not it is an "external routine". The Purpose of the Main Program module is listed as well.

PS 620141213
1 November 1985

PARSE NDML COMMAND Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
NDMLPAR	Purpose-->**** PURPOSE NOT FOUND BY STRIPPER ****	
	INITCMD	External routine
	NDMLPAS	External routine
	YYINPUT	External routine
	YYOUTPUT	External routine

PS 620141213
1 November 1985

PARSE NDML COMMAND Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
YYPARSE	Purpose--,**** PURPOSE NOT FOUND BY STRIPPER ****	
	ADDNCORR	External routine
	ADD_CORR	External routine
	ADD_TO_CNT	External routine
	ADD_TO_LST	External routine
	PRINTF	External routine
	SPRINTF	External routine
	STRLEN	External routine
	WRITERR	External routine
	YYERROR	Well-defined module
	YYLEX	External routine

3.10.8 Module Documentation

The following documentation describes information which is specific to each individual module being documented in this specification as listed in section 3.10.2. It provides a compact way of getting information that would be otherwise buried within each module's source code.

The specific items in this module documentation have the following meanings:

NAME:	Name of program Module.
PURPOSE:	Purpose of Module as detailed in the source code.
LANGUAGE:	Programming language source code is written in. The choices are: VAX-11 FORTRAN C (I/S-1 Workbench 'C') VAX-11 COBOL
MODULE TYPE:	Whether a Program, Subroutine, or Function.
SOURCE FILE:	Name of Source File from file specification.
SOURCE FILE TYPE:	Source File Extension from file specification.
HOST:	Whether this is a host-dependent routine (VAX or IBM) or blank if host-independent.
SUBSYSTEM:	IISS sub-system this file resides in.
SUBDIRECTORY:	Sub-directory of that subsystem in which this file resides.
DOCUMENTATION GROUP:	Name of documentation group of which this source file is a member.
DESCRIPTION:	A description of the module as

PS 620141213
1 November 1985

obtained from the source code.

ARGUMENTS:

The arguments with which this routine is called if it is a Subroutine or a Function.

INCLUDE FILES:

A list of all the files that are included into this module as well as their purposes.

ROUTINES CALLED:

Subroutines or Functions, either documented or external, called by this module, if any.

CALLED DIRECTLY BY:

The documented routines which call this module, if any.

USED IN MAIN PROGRAM(S):

The documented Main Programs which contain this module in their parts list according to the list in section 3.10.7.

The Module Documentation is arranged alphabetically according to Module Name.

PS 620141213
1 November 1985

PARSE NDML COMMAND Module Documentation

NAME: NDMLPAR
PURPOSE: **** PURPOSE NOT FOUND BY STRIPPER ****
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: NDMLPAR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41213

DESCRIPTION:

ARGUMENTS:

COM NO = INT *
STATUS = INT *

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

INITCMD
NDMLPAS
YYINPUT
YYOUTPUT

PS 620141213
1 November 1985

PARSE NDML COMMAND Module Documentation

NAME: YYERROR
PURPOSE: **** PURPOSE NOT FOUND BY STRIPPER ****
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: NDMLYTB
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41213

DESCRIPTION:

ARGUMENTS:

S = CHAR *

INCLUDE FILES:

NDMLYAC.IN	- **** PURPOSE NOT FOUND BY STRIPPER ****
STDIO	- **** PURPOSE NOT FOUND BY STRIPPER ****
CMDS	- COMMAND NO. FOR EACH NDDL/NDML COMMAND
LISTS	- PROVIDES THE DIMENSIONS OF THE NDDL
	LISTS
LISTID	- PROVIDES LIST OF PARSED OBJECTS
NDMLEXY	- LEX GENERATED INCLUDE FILE

ROUTINES CALLED:

SPRINTF
STRLEN
WRITERR

CALLED DIRECTLY BY:

YYPARSE - **** PURPOSE NOT FOUND BY STRIPPER ****

PS 620141213
1 November 1985

USED IN MAIN PROGRAM(S):

YYPARSE

- **** PURPOSE NOT FOUND BY STRIPPER ****

PS 620141213
1 November 1985

PARSE NDML COMMAND Module Documentation

NAME: YYPARSE
PURPOSE: **** PURPOSE NOT FOUND BY STRIPPER ****
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: NDMLYTB
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41213

DESCRIPTION:

INCLUDE FILES:

NDMLYAC.IN	- **** PURPOSE NOT FOUND BY STRIPPER ****
STDIO	- **** PURPOSE NOT FOUND BY STRIPPER ****
CMDS	- COMMAND NO. FOR EACH NDDL/NDML COMMAND
LISTS	- PROVIDES THE DIMENSIONS OF THE NDDL LISTS
LISTID	- PROVIDES LIST OF PARSED OBJECTS
NDMLEXY	- LEX GENERATED INCLUDE FILE

ROUTINES CALLED:

PRINTF	
ADD_TO_LST	
ADD_CORR	
STRLEN	
ADDNCORR	
ADD_TO_CNT	
YYERROR	
YYLEX	- **** PURPOSE NOT FOUND BY STRIPPER ****

PS 620141213
1 November 1985

PARSE NDML COMMAND Module Documentation

NAME: YYWRAP
PURPOSE: **** PURPOSE NOT FOUND BY STRIPPER ****
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: NDMLYTB
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41213

DESCRIPTION:

INCLUDE FILES:

NDMLYAC.IN - **** PURPOSE NOT FOUND BY STRIPPER ****
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CMDS - COMMAND NO. FOR EACH NDDL/NDML COMMAND
LISTS - PROVIDES THE DIMENSIONS OF THE NDDL
LISTS
LISTID - PROVIDES LIST OF PARSED OBJECTS
NDMLEXY - LEX GENERATED INCLUDE FILE

PS 620141213
1 November 1985

3.10.9 Include File Descriptions

The following list contains a purpose and description of each include file listed in 3.10.4 as specified in the source code. The language it is written in is also given.

PS 620141213
1 November 1985

PARSE NDML COMMAND Include File Description

FILE NAME: CMDS
PURPOSE: COMMAND NO. FOR EACH NDDL/NDML COMMAND
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
CONTAINS THE UNIQUE COMMAND NUMBER FOR EACH NDDL AND NDML
COMMAND. THIS NUMBER IS USED IN PROCESSING TO DETERMINE THE
APPROPRIATE PROCESSOR TO EXECUTE.

PS 620141213
1 November 1985

PARSE NDML COMMAND Include File Description

FILE NAME:	LISTID
PURPOSE:	PROVIDES LIST OF PARSED OBJECTS
LANGUAGE:	C

DESCRIPTION:

DESCRIPTION

PS 620141213
1 November 1985

PARSE NDML COMMAND Include File Description

FILE NAME:	LISTS
PURPOSE:	PROVIDES THE DIMENSIONS OF THE NDDL LISTS
LANGUAGE:	C

DESCRIPTION:

DESCRIPTION

PS 620141213
1 November 1985

PARSE NDML COMMAND Include File Description

FILE NAME: NDMLEXY
PURPOSE: LEX GENERATED INCLUDE FILE
LANGUAGE: C

DESCRIPTION:

DESCRIPTION

PARSE NDML COMMAND Include File Description

FILE NAME: STD TYP
PURPOSE: STANDARD TYPE DEFINITIONS
LANGUAGE: C

DESCRIPTION:

DESCRIPTION

THIS FILE ENSURES THAT THE FOLLOWING STANDARD TYPES ARE
AVAILABLE:

FLOAT	- SINGLE PRECISION FLOAT
DOUBLE	- DOUBLE PRECISION FLOAT
LONG	- 32 BIT (OR LARGER) SIGNED INTEGER
LBITS	- 32 BITS (OR MORE) FOR BIT MANIPULATION
INT	- NATURAL SIZE SIGNED INTEGER
UNSIGNED	- NATURAL SIZE UNSIGNED INTEGER
BOOL	- NATURAL SIZE LOGICAL (ZERO / NON-ZERO
ONLY)	
SHORT	- 16 BIT (OR LARGER) SIGNED INTEGER
USHORT	- 16 BIT (OR LARGER) UNSIGNED INTEGER
BITS	- 16 BITS (OR MORE) FOR BIT MANIPULATION
CHAR	- SINGLE MACHINE CHARACTER (REAL CHARACTERS ALWAYS POSITIVE)
TINY	- 8 BIT (OR LARGER) SIGNED INTEGER
UTINY	- 8 BIT (OR LARGER) UNSIGNED INTEGER
TBITS	- 8 BITS (OR MORE) FOR BIT MANIPULATION
TBOOL	- 8 BIT (OR LARGER) LOGICAL (ZERO / NON-ZERO
ONLY)	
METACHAR	- 16 BIT (OR LARGER) AUGMENTED CHARACTER
(SIGNED)	
VOID	- FUNCTION THAT RETURNS NO VALUE
FORTRAN	- STORAGE CLASS FOR FOREIGN (NON-C) ROUTINES OR C ROUTINES WHICH ARE CALLABLE FROM FOREIGN ROUTINES
	 SINCE NOT ALL COMPILERS SUPPORT USHORT,

PS 620141213
1 November 1985

TINY, AND UTINY,
THE FUNCTIONS
USHORT(), TINY(), AND UTINY() SHOULD BE USED
WHENEVER REFERENCING THEM.

IN ADDITION, THE FOLLOWING UTILITY MACROS
ARE DEFINED: LURSHIFT(N, B)
- UNSIGNED LONG RIGHT SHIFT MAX(A, B)
- MAXIMUM OF A AND B MIN(A, B) - MINIMUM OF
A AND B

PARSE NDML COMMAND Include File Description

ABS(A) - ABSOLUTE VALUE OF A
STRASN(A, B) - TRANSPORTABLE A - B FOR STRUCTURES
NULL - NULL POINTER VALUE (0)
TRUE - 1
FALSE - 0
SUCCESS - EXIT(SUCCESS) INDICATES SUCCESSFUL
COMPLETION
FAILURE - EXIT(FAILURE) INDICATES ERRORS

**THE FOLLOWING SYMBOLS SHOULD BE DEFINED BASED ON THE
COMPILER BEING USED:**

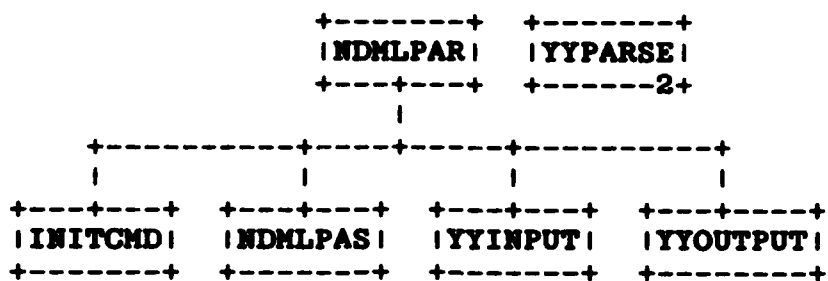
USHORT - COMPILER SUPPORTS UNSIGNED SHORT
TINY - COMPILER TREATS CHAR AS SIGNED
UTINY - CHAR IS SIGNED AND COMPILER SUPPORTS
UNSIGNED CHAR
VOID - COMPILER SUPPORTS VOID
FORTRAN - COMPILER SUPPORTS FORTRAN
STRASN - DEFINE APPROPRIATE MACRO
SUCCESS - DEFINE APPROPRIATE VALUE IF NOT 0
FAILURE - DEFINE APPROPRIATE VALUE IF NOT 1

3.10.10 Hierarchy Chart

The following hierarchy charts show the relationships between all of the modules mentioned in the above documentation. A module may call a subroutine several times within its code, but the call will only be shown once as a single relationship on this hierarchy chart. All modules shown at the top of the first page are considered Main Programs as described in section 3.10.1 above.

There is an internal paging scheme as marked by the numbers in the upper right corner of each page. An index after the last page of the chart shows where a routine and its calls are first defined. If a routine has no page reference, it either makes no calls or is an external routine. A continuation box on the end of a tree limb shows where that the tree continues on the page numbered mentioned. A number in a box with a routine name points to the page where the routine is further defined within the hierarchy tree. If there is no number in a box, the routine either makes no calls or is an external routine.

1



PS 620141213
1 November 1985

2

```
      +-----+
      |YYPARSE|
      +-----+
      |
+-----+-----+-----+-----+
|         |         |         |         |
+-----+ +-----+ +-----+ +-----+ +-----+
|PRINTF| |ADD_TO_LST| |ADD_CORR| |STRLEN| |(CONT)|
+-----+ +-----+ +-----+ +-----+ +-----3+
```



PS 620141213
1 November 1985

ADDNCORR
ADD_CORR
ADD_TO_CNT
ADD_TO_LST
INITCMD
NDMLPAR.....1
NDMLPAS
PRINTF
SPRINTF
STRLEN
WRITERR
YYERROR.....3
YYINPUT
YYLEX
YYOUTPUT
YYPARSE.....2

2
PS 620141213
1 November 1985

3.11 Program Listings Comments

This information is contained in the Module Descriptions in section 3.10.

SECTION 4

QUALITY ASSURANCE PROVISIONS

4.1 Introduction and Definitions

"Testing" is a systematic process that may be preplanned and explicitly stated. Test techniques and procedures may be defined in advance, and a sequence of test steps may be specified. "Debugging" is the process of isolation and correction of the cause of an error.

"Antibugging" is defined as the philosophy of writing programs in such a way as to make bugs less likely to occur and when they do occur, to make them more noticeable to the programmer and the user. In other words, as much error checking as is practical and possible in each routine should be performed.

4.2 Computer Programming Test and Evaluation

The quality assurance provisions for test consists of the normal testing techniques that are accomplished during the construction process. They consist of design and code walk-throughs, unit testing, and integration testing. These tests are performed by the design team. Structured design, design walk-through and the incorporation of "antibugging" facilitate this testing by exposing and addressing problem areas before they become coded "bugs."

END

7-87

DTIC